IN THE CLAIMS:

1. (original) An error correction code encoder for encoding data in accordance with one or more factors of a generator polynomial $g(x) = g_1(x)*g_2(x)$, the encoder including:
   A. a first stage for selectively multiplying symbols by $g_1(x)$ to produce a product or dividing the symbols by $g_1(x)$ to produce one or both of a quotient $q_1(x)$ and a remainder $r_1(x)$;
   B. a second stage for dividing $q_1(x)$ by the polynomial $g_2(x)$ to produce a remainder $r_2(x)$;
   C. a controller for operating the first and second stages, the controller operating
      a. in a first mode to supply the data to the first stage, the associated quotient $q_1(x)$ to the second stage, the associated remainder $r_2(x)$ produced by the second stage back to the first stage and control the first stage to produce the product $r_2(x)*g_1(x)$;
      b. in a second mode to by-pass the second stage; and
      c. in a third mode to pass the data to the second stage as the quotient $q_1(x)$; and
   D. a processor for producing ECC symbols by manipulating the remainders and products produced by the first and second stages.

2. (original) The encoder of claim 1 wherein the controller operating in the third mode passes the data through the first stage to supply the data to the second stage as the quotient $q_1(x)$.

3. (original) The encoder of claim 1 wherein the second stage includes

j stages that multiply the symbols by coefficients of degree-one factors of $g_2(x)$; and

a multiplexer that selectively operates a stage j-i as the last stage, where $0 \leq i < j$.

4. (original) The encoder of claim 1 wherein the second stage encodes in accordance with $g_2(x) = g_3(x)*g_4(x)$, the second stage including:

a first sub-stage for selectively multiplying symbols by $g_3(x)$ to produce a product or dividing the symbols by $g_3(x)$ to produce one or both of a quotient $q_3(x)$ and a remainder $r_3(x)$;

E. a second sub-stage for dividing $q_3(x)$ by the polynomial $g_4(x)$ to produce a remainder $r_4(x)$;

F. a controller for operating the first and second sub-stages, the controller operating

in a first mode to supply the quotient $q_1(x)$ to the first sub-stage, the associated quotient $q_3(x)$ to the second stage, the associated remainder $r_4(x)$ produced by the second sub-stage back to the first sub-stage and control the first sub-stage to produce the product $r_4(x)*g_3(x)$;

in a second mode to by-pass the second sub-stage; and

in a third mode to pass the quotient $q_1(x)$ to the second sub-stage as the quotient $q_3(x)$;

wherein the second stage provides to the processor the remainders and products produced by the first and second sub-stages.

5. (original) An error correction code encoder for encoding data in accordance with one or more factors of a generator polynomial $g(x) = g_1(x)*g_2(x)$, the encoder including:

A. a first stage for selectively multiplying the symbols by $g_1(x)$ to produce a product or dividing symbols by $g_1(x)$ to produce one or both of a quotient $q_1(x)$ and a remainder $r_1(x)$;

B. a second stage for dividing $q_1(x)$ by one or more factors of the polynomial $g_2(x)$ to produce a remainder $r_2(x)$;

C. a controller for operating the first and second stages, the controller operating

a. in a first mode to supply the data to the first stage, the associated quotient $q_1(x)$ to the second stage, the associated remainder $r_2(x)$ produced by the second stage back to the first stage and control the first stage to produce the product $r_2(x)*g_1(x)$;

b. in a second mode to by-pass the second stage; and

c. in a third mode to pass the data to the second stage as the quotient $q_1(x)$; and

D. a processor for producing ECC symbols by manipulating the remainders and products produced by the first and second stages.

6. (original) The encoder of claim 4 wherein the second stage includes

j stages that multiply the symbols by coefficients of degree-one factors of $g_2(x)$; and

a multiplexer that selectively operates a stage j-i as the last stage, where $0 \leq i < j$.

7. (currently amended) A method for encoding k data symbols in accordance with one or more factors of a generator polynomial $g(x) = g_1(x)*g_2(x)* \ldots *g_f(x)$ of degree n-k, the method including:

A. using one or more factors of g(x) as a selected polynomial p(x) of degree m, where $1 \leq m \leq n-k$;

B. dividing the data symbols by a first factor $p_1(x)$ of p(x) to produce <u>one or both of</u> a remainder $r_1(x)$ and/~~or~~ a quotient $q_1(x)$, the first factor having degree s;

C. if p(x) has more factors dividing the quotient $q_1(x)$ by a next factor $p_i(x)$ of the polynomial p(x) to produce a remainder $r_i(x)$;

D. if p(x) has more factors dividing the quotient $q_i(x)$ by a next factor $p_{i+1}(x)$ to produce a remainder $r_{i+1}(x)$ and/or a quotient $q_{i+1}(x)$;

E. repeating steps C and D for the remaining factors of p(x); and

F. manipulating the remainders to produce redundancy symbols <u>that are the ECC symbols for detecting and correcting errors in the k data symbols</u>.

8. (original) The method of claim 7 wherein the step of manipulating the remainders includes the steps of

multiplying the respective remainders r_i by associated factors $p_t(x)$, for t = 1, 2,..., i-1;

adding the results to $r_1(x)$ to produce a remainder sum; and

shifting the remainder sum by x^{n-s} to produce ECC symbols.

9. (currently amended) A method for encoding k data symbols in accordance with one or more factors of a generator polynomial $g(x) = g_1(x)*g_2(x)$ of degree n-k, the method including:

A. selecting $g_1(x)$, $g_2(x)$ or $g_1(x)*g_2(x)$ as a polynomial p(x) of degree m, where $1 \leq m \leq n-k$;

B. dividing the data symbols by a first factor $p_1(x)$ of p(x) to produce <u>one or both of</u> a remainder $r_1(x)$ and~~/or~~ a quotient $q_1(x)$, the first factor having degree s;

C. if p(x) has a second factor dividing the quotient $q_1(x)$ by a next factor $p_2(x)$ of the polynomial p(x) to produce a remainder $r_2(x)$; and

D. manipulating the remainders to produce redundancy symbols <u>that are the ECC symbols for detecting and correcting errors in the k data symbols</u>.

10. (original) The method of claim 9 wherein the step of manipulating the remainders includes using $r_1(x)$ as the ECC symbols.

11. (original) The method of claim 10 wherein the step of manipulating the remainders includes the steps of

multiplying $r_2(x)$ by $p_1(x)$ to produce a product,

adding the product to $r_1(x)$ and

shifting the result by x^{n-s}.

12. (original) A decoder for decoding a code word that is encoded in accordance with one or more factors of a generator polynomial $g(x) = g_1(x)*g_2(x)$, the decoder including:

A. a first stage for selectively multiplying the symbols by $g_1(x)$ or dividing symbols by $g_1(x)$ to produce either a remainder $r_1(x)$, a quotient $q_1(x)$ or both the remainder and the quotient;

B. a second stage for dividing the quotient $q_1(x)$ by the polynomial $g_2(x)$ to produce a remainder $r_2(x)$;

C. a controller for operating the first and second stages, the controller operating

a. in a first mode to supply the data to the first stage, the associated quotient $q_1(x)$ to the second stage, the associated remainder $r_2(x)$ produced by the second stage back to the first stage and control the first stage to produce the product $r_2(x)*g_1(x)$;

b. in a second mode to by-pass the second stage; and

c. in a third mode to pass the data to the second stage as the quotient $q_1(x)$; and

D. a processor for producing ECC symbols by manipulating the remainders and products produced by the first and second stages, the processor comparing the ECC symbols with the code word ECC symbols and, as necessary, producing error syndromes and correcting errors in the data to produce error-free data.

13. (original) A decoder for decoding code words encoded in accordance with one or more factors of a generator polynomial $g(x) = g_1(x)*g_2(x)$, the decoder including:

A. a first stage for selectively dividing symbols by $g_1(x)$ to produce a quotient $q_1(x)$ and/or a remainder $r_1(x)$ or multiplying the symbols by $g_1(x)$ to produce a product;

B. a second stage for dividing $q_1(x)$ by one or more factors of the polynomial $g_2(x)$ to produce a remainder $r_2(x)$ or producing error syndromes associated with the one or more factors of $g_2(x)$;

C. a controller for operating the first and second stages, the controller operating

a. in a first mode to supply the data to the first stage, the associated quotient $q_1(x)$ to the second stage, the associated remainder $r_2(x)$ produced by the second stage back to the first stage and control the first stage to produce the product $r_2(x)*g_1(x)$;
b. in a second mode to by-pass the second stage; and
c. in a third mode to pass the data to the second stage as the quotient $q_1(x)$; and
d. in a fourth mode to operate the second stage to produce error syndromes associated with the one or more factors of $g_2(x)$; and

D. a first processor that produces ECC symbols by manipulating the remainders and products produced by the first and second stages; and

E. a second processor that produces error syndromes associated with $g_1(x)$ and uses the error syndromes produced by the second stage to, as necessary, correct errors in the data and produce error-free data.

14. (new) The method of claim 7 further including the step of producing a data code word by concatenating the ECC symbols to the k data symbols.

15. (new) The method of claim 9 further including the step of producing a data code word by concatenating the ECC symbols to the k data symbols.

16. (new) The method of claim 7 further including the step of comparing the ECC symbols produced in step F with the ECC symbols of a code word containing the k data symbols and, as necessary, producing error syndromes and correcting errors in the data symbols to produce error-free data.

17. (new) The method of claim 8 further including the step of comparing the ECC symbols produced in step F with the ECC symbols of a code word containing the k data symbols and,

as necessary, producing error syndromes and correcting errors in the k data symbols to produce error-free data.

18. (new) The method of claim 9 further including the step of comparing the ECC symbols produced in step D with the ECC symbols of a code word containing the k data symbols and, as necessary, producing error syndromes and correcting errors in the k data symbols to produce error-free data.

19 (new) The method of claim 10 further including the step of comparing the ECC symbols produced in step D with the ECC symbols of a code word containing the k data symbols and, as necessary, producing error syndromes and correcting errors in the k data symbols to produce error-free data.

20. (new) The method of claim 11 further including the step of comparing the ECC symbols produced in step D with the ECC symbols of a code word containing the k data symbols and, as necessary, producing error syndromes and correcting errors in the k data symbols to produce error-free data.